Appendix

Definitions in Support of Applicant's Arguments

OA Short for *office automation*

object Generally, any item that can be individually selected and manipulated. This can include shapes and pictures that appear on a display screen as well as less tangible software entities. In object-oriented programming, for example, an object is a self-contained entity that consists of both data and procedures to manipulate the data.

⇒ See also BLOB; CORBA; OBJECT ORIENTED; OBJECT-ORIENTED GRAPHICS; OBJECT-ORIENTED PROGRAMMING; OLE; OMG.

object code The code produced by a compiler. Programmers write programs in a form called source code. The source code consists of instructions in a particular language, like C or FORTRAN. Computers, however, can execute only instructions written in a low-level language called *machine language*.

To get from source code to machine language, the programs must be transformed by a compiler. The compiler produces an intermediary form called object code. Object code is often the same as or similar to a computer's machine language. The final step in producing an executable program is to transform the object code into machine language, if it is not already in this form. This can be done by a number of different types of programs, called *assemblers, binders, linkers,* and *loaders.*

⇒ See also ASSEMBLER; ASSEMBLY LANGUAGE; CODE; COMPILE; LIBRARY; LINK; LOAD; MACHINE LANGUAGE.

Object Linking and Embedding See OLE.

Object Management Group See OMG.

object oriented A popular buzzword that can mean different things depending on how it is used. *Object-oriented programming (OOP)* refers to a special type of programming that combines data structures with functions to create reusable objects (see under *object-oriented programming).* Object-oriented graphics is the same as *vector graphics.*

Otherwise, the term *object-oriented* is generally used to describe a system that deals primarily with different types of objects, and where the actions you can take depend on what type of object you are manipulating. For example an object-oriented draw program might enable you to draw many types of objects, such as circles, rectangles, triangles, and so on. Applying the same action to each of these objects, however, would

produce different results. If the action is *Make 3D,* for instance, the result would be a sphere, box, and pyramid, respectively.

⇒ See also OBJECT-ORIENTED PROGRAMMING; SMALLTALK; VECTOR GRAPHICS.

object-oriented graphics The representation of graphical objects, such as lines, arcs, circles, and rectangles, with mathematical formulas. This method of describing objects enables the system to manipulate the objects more freely. In an object-oriented system, for example, you can overlap objects but still access them individually, which is difficult in a bit-mapped system. Also, object-oriented images profit from high-quality output devices. The higher the resolution of a monitor or printer, the sharper an object-oriented image will look. In contrast, bit-mapped images always appear the same regardless of a device's resolution.

One of the most widely used formats for object-oriented graphics is Postscript. Postscript is a page description language (PDL) that makes it possible to describe objects and manipulate them in various ways. For example, you can make objects smaller or larger, turn them at various angles, and change their shading and color. A font described in PostScript, therefore, can easily be transformed into another font by changing its size or weight. Object-oriented fonts are called *outline fonts, scalable fonts,* or *vector fonts.*

Object-oriented graphics is also called *vector graphics,* whereas bit-mapped graphics is sometimes called *raster graphics.*

⇒ See also BIT-MAPPED GRAPHICS; GRAPHICS; POSTSCRIPT; SCALABLE FONT; VECTOR GRAPHICS.

object-oriented programming A type of programming in which programmers define not only the data type of a data structure but also the types of operations (functions) that can be applied to the data structure. In this way, the data structure becomes an *object* that includes both data and functions. In addition, programmers can create relationships between one object and another. For example, objects can *inherit* characteristics from other objects.

One of the principal advantages of object-oriented programming techniques over procedural programming techniques is that they enable programmers to create modules that do not need to be changed when a new type of object is added. A programmer can simply create a new object that inherits many of its features from existing objects. This makes object-oriented programs easier to modify.

To perform object-oriented programming, one needs an *object-oriented programming language (OOPL).* C++ and Smalltalk are two of the more popular languages, and there are also object-oriented versions of Pascal.

⇒ See also C++; CLASS; COMPONENT SOFTWARE; DISTRIBUTED COMPUTING; EIFFEL; ENCAPSULATION; JAVA; OBJECT ORIENTED; OMG; OVERLOADING; POLYMORPHISM; SMALLTALK; UML; VISUAL C++.

Object Request Broker See ORB.

21

22

searchSMB.com Definitions - powered by whatis.com

instantiation

powered by whatis?com

In programming, instantiation is the creation of a real *instance* or particular realization of an abstraction or template such as a class of objects or a computer process. To instantiate is to create such an instance by, for example, defining one particular variation of object within a class, giving it a name, and locating it in some physical place.

1) In object-oriented programming, some writers say that you instantiate a class to create an object, a concrete instance of the class. The object is an executable file that you can run in a computer.

2) In the object-oriented programming language, Java, the object that you instantiate from a class is, confusingly enough, called a class instead of an object. In other words, using Java, you instantiate a class to create a specific class that is also an executable file you can run in a computer.

3) In approaches to data modeling and programming prior to object-oriented programming, one usage of *instantiate* was to make a real (data-filled) object from an abstract object as you would do by creating an entry in a database table (which, when empty, can be thought of as a kind of class template for the objects to be filled in).

>> Find white papers, products and vendors related to instantiation.

Read more about it:

>> SearchVB.com has resources for this and other Visual Basic programming terms.

23

MENU
Home
Term of the Day
New Terms
Pronunciation
New Links
Quick Reference
Did You Know?
Search Tool
Tech Support
Webopedia Jobs
About Us
Link to Us
Advertising

Compare Prices: go
HardwareCentral

Talk To Us...
Submit a URL
Suggest a Term
Report an Error

internet.com
Developer
Downloads
International
Internet Lists
Internet News
Internet Resources
IT
Linux/Open Source
Personal Technology
Small Business
Windows Technology
xSP Resources

Search internet.com
Advertise
Corporate Info
Newsletters
Tech Jobs
E-mail Offers

internet commerce

invoke

Last modified: Sunday, September 01, 1996

To activate. One usually speaks of *invoking* a function or routine in a program. In this sense, the term *invoke* is synonymous with *call*.

•E-mail this definition to a colleague•

*For internet.com pages about **invoke** **CLICK HERE**. Also check out the following links!*

LINKS = Great Page!

Related Categories

Procedures, Functions and Routines

Related Terms

call

function

routine

(Webopedia)
Give Us Your Feedback

24

Michael Barr's

Embedded Systems Glossary

Buy the book, which defines over 2,800 commonly used terms.

Last Update: 18 June 2003

[# A B C D E F G H I J K L M N O P Q R S T U V W X Y Z]

[Suggest a Change or a New Term]

R

race condition

1. *n.* A situation in which the combined effects of two or more programmatic threads (or a single thread and an ISR) varies depending on the precise order in which the instructions of each are executed.

Race conditions can be eliminated by surrounding critical code sections that must be executed without interruption with a pair of mutex take and release system calls. To prevent race conditions involving ISRs, interrupts must be disabled for the duration of the critical section.

EXAMPLE: If two threads both try to increment a shared global variable (`x = x + 1;`) and they race, the result of both threads incrementing the variable once from an initial value (`x = 0;`) can be either 2 (correct) or 1 (incorrect).

This race condition exists if either increment step is not executed atomically. If a context switch occurs in the middle of an increment (which is typically a sequence of three CPU instructions to read the old value into a register, increment the content of the register, then write the new value), an incorrect value can result. The error might not always occur, making tracking down such bugs incredibly difficult.

The outcome (final value of x) in this case is dependent on the precise order in which the instructions of the two threads are executed. The shared data and random nature of preemptive context switches are the culprits that cause the race condition.

2. *n.* A situation in digital logic where timing errors cause erratic outputs.

random access memory

n. A broad classification of memory devices that includes all devices in which individual memory locations can be read or written in any order required by the application. Abbreviated

25

RAM. Misused to mean memory that can be both read and written, but the term is so broadly (mis)used in this fashion that nearly everyone assumes random access is the same as read-write. *See also* read-only memory. [more]

rate monotonic analysis

n. The process of analyzing a real-time system to assign individual thread priorities according to the rate monotonic algorithm. [more]

read-only memory

n. A broad classification of memory devices that includes all devices in which memory locations cannot be modified. Abbreviated ROM. Misused to mean any nonvolatile memory, including flash and EEPROM, that can be modified in-system. *See also* random access memory. [more]

real-time operating system

n. An operating system designed specifically for use in real-time systems. Abbreviated RTOS. [more]

real-time system

n. Any computer system, embedded or otherwise, that has timeliness requirements. The following question can be used to distinguish real-time systems from the rest: "Is a late answer as bad, or even worse, than a wrong answer?" In other words, what happens if the computation doesn't finish in time? If nothing bad happens, it's not a real-time system. If someone dies or the mission fails, it's generally considered "hard" real-time, which is meant to imply that the system has hard deadlines. Everything in between is "soft" real-time. [more]

EXAMPLE: Most industrial automation equipment has deadlines. If the bottle doesn't get a cap applied properly as it passes by on the production line, there is a failure. However, the consequences of that failure would not be as severe as the consequences of a failure in an airplane, a pacemaker, or any of a thousand other hard real-time systems.

recursive

adj. Said of software that calls itself. Recursion should generally be avoided in an embedded system, since it frequently requires a large stack.

reentrant

adj. Said of software that can be executed multiple times simultaneously. A reentrant function can be safely called recursively or from multiple tasks. The key to making code reentrant is to ensure mutual exclusion whenever accessing global variables or shared registers. [more]

register

n. A memory-like location that is part of a processor or an I/O device. The reference to the register is encoded as part of the instruction, not as a discrete address. A processor register is much faster to read or write than a location in memory. Generally, each bit or set of bits within a peripheral register controls or tracks some behavior of the larger device.

relocatable

26

Michael Barr's

Embedded Systems Glossary

Buy the book, which defines over 2,800 commonly used terms.

Last Update: 18 June 2003

[# A B C D E F G H I J K L M N O P Q R S T U V W X Y Z]

[Suggest a Change or a New Term]

M

MAC

(like the truck) 1. *abbr. See* multiply-and-accumulate.

2. *abbr. See* MAC address.

MAC address

n. A globally unique 48-bit hardware address assigned to each device on a network. Every system on a physical network, like Ethernet or Token Ring, includes a peripheral called a network controller. This chip is the processor's interface to the physical communications medium. As part of its initialization, the network controller must be fed a unique hardware address to use when communicating over the network. In the case of Ethernet, the hardware address is a 48-bit value. To guarantee global uniqueness, the upper 24 bits are controlled by the IEEE, which allocates them to individual device manufacturers. See OUI for more information about obtaining a block of Ethernet addresses for your company. [more]

mebi-

(meh bee) *pre.* The prefix meaning 2**20. Abbreviated Mi.

* 1 mebibit: 1 Mibit = 1,048,576 bits
* 1 mebibyte: 1 MiB = 1,048,576 bytes

See also binary prefixes, mega-.

memory map

n. A table or diagram containing the name and address range of each peripheral and memory device within a processor's memory space. Memory maps are a helpful aid in getting to know one's target.

17

Motor Industry Software Reliability Association, April 1998. Available for purchase at http://www.misra.org.uk.

monitor

1. *n.* A language-level intertask synchronization primitive. Java is the only language in the embedded systems space that supports monitors.

2. *n.* The CRT or LCD display attached to a computer.

3. *See* debug monitor.

multiply-and-accumulate

adj. Describes a special CPU instruction, common on digital signal processors, that performs both a multiplication and an addition in a single instruction cycle. The result of the multiplication is typically added to a sum kept in a register. Abbreviated MAC. A multiply-and-accumulate instruction is helpful for speeding up the execution of the many digital filters and transforms required in signal processing applications. In recent years, many microprocessor and microcontroller makers have included a MAC instruction on their products as well.

multiprocessing

n. The use of more than one processor in a single computer system. So-called multiprocessor systems usually have a common memory space through which all of the processors can communicate and share data. In addition, some multiprocessor systems support parallel processing.

multitasking

n. The execution of multiple software routines in pseudoparallel. Each routine represents a separate thread of execution. The operating system is responsible for simulating parallelism by parceling out the processor's time to the individual threads. [more]

multithreading

See multitasking.

mutex

(mew tex) *n.* An operating system data structure used by tasks to ensure exclusive access to shared variables or hardware registers. Short for mutual exclusion. A mutex is a multitasking-aware binary flag that can be used to synchronize the activities of multiple tasks. As such, it can protect critical sections from interruption and shared resources from simultaneous accesses.

USAGE: The term "mutex" is best reserved only for binary semaphores that are aware of the potential for priority inversions and implement an appropriate workaround. But beware that many RTOS vendors do not make such distinctions.

mutual exclusion

n. A guarantee of exclusive access to a shared resource. In embedded systems, the shared resource is typically a block of memory, a global variable, a peripheral, or a set of registers. Mutual exclusion is typically achieved with the use of a mutex. 28

Though originally based on the glossary in the book Programming Embedded Systems in C and C++, this online version has grown considerably and is a living reference. In its present form it represents a subset of the much larger print Embedded Systems Dictionary. If you wish to cite it in your work, you may find the following MLA-style information helpful:

Barr, Michael. "Embedded Systems Glossary." Online at http://www.netrino.com/Publications/Glossary/. June 2003.